Chapter 8

Generics

Syntax of Generics Class:-

Class class\_name<type\_param> { }

Instantiation of Generics Class:-

Class\_name<type\_arg\_list> varname=new Class\_name<type\_arg\_list> ();

Example of Generics:-

Public class Tuple<X,Y>

{

Private X x;

Private Y y;

Public Tuple(X x,Y y)

{

This.x=x;

This.y=y;

}

}

And we can store Tuple to another Tuple,

Tuple<String, Tuple<Integer,Integer>> tuple =new Tuple<String,Tuple<Integer,Integer>> ();

Diamond Operator:-

Map<String,List<String>> ar=new HashMap<String,List<String<String>>();

Generics and subtyping:-

Rules:-

If a class Foo subclass of Bar, Generics of Foo not subclass of Generics of Bar class.

Bounded Types:-

আমরা এর মাধ্যমে জেনেরিক এর টাইপ নির্ধারন করে দিতে পারি।

যেমন, এই ক্লাস টি টাইপ প্যারামিটার হিসাবে শুধুমাত্র নম্বার নিবে নতুবা কাজ করবে না।

Ex:-

Class State<T extends number>

এখন এই ক্লাস এর অবজেক্ট শুধুমাত্র Double, Float, Long, Short, BigInteger, BigDecimal, Byte হবে।

Wildcard Arguments:-

যখন প্লেস হোল্ডারের টাইপ আমাদের জানানেই তখন Wildcard ‘?’ operator ব্যবহৃত হয়।

বাউন্ডেড টাইপ এর মতো আমরা ওয়াইল্ডকার্ড আর্গুমেন্ট বাউন্ডেড ব্যবহৃত হয়।

Ex:-

Public void process(List <? Extends Foo>list)

{

}

এটি শুধুমাত্র Foo এর সাব ক্লাস গুলোকে প্রসেস করতে পারবে। এটি Upper Bounded wildcard

Public void addNumber(List< ? super Integer> list)

{

}

এটি Lower Bounded wildcard

Why we use:-

ধরি একটি মেথড যা List এর value point করবে।

Public void print(List<Object> ls)

{

For(Object o:ls)

System.out.println(o);

}

উপরের মেথড টি শুধু List<Object> গ্রহন করবে List<String> বা List<Integer> করবে না। এই সমস্যা দূর করার জন্য এটি ব্যবহৃত হয়।

Public void print(List<?> ls)

{

For(Object o:ls)

{

System.out.println(o);

}

}

Generics Method:-

Public <K,V> Boolean isSet(K i, V j)

{

Return i%j==0;

}

Type Erasure

আমরা Generics type লিখে Type টি নিশ্চিত করতে পারি। জেনেরিক ব্যবহার করে যে আমরা যে কোড লিখি তা শুধু কম্পাইল টাইম এর জন্য, ইনফরমেশন গুলো মুছে ফেলা হয়। একে Type Erasure বলে।

Rules of Generics:-

1. Generics Type এর Instance তৈরি করা যায় না।
2. Generics Type এর Array তৈরি করা যায়না ।

তবে,

T[] object=(T[]) new Object[capacity];

এর মাধ্যমে তৈরি করা যায় না।

1. Static context এর Generics type parameter করা যায় না।
2. জাভাতে Exception ক্লাস গুলো Generics হতে পারে না।